Journal of King Saud University – Computer and Information Sciences 35 (2023) 393–404

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Local generalization and bucketization technique for personalized privacy preservation

Boyu Li, Kun He [*]

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

ARTICLE INFO

Article history:
Received 1 November 2022
Revised 17 November 2022
Accepted 14 December 2022
Available online 28 December 2022

Keywords:
Privacy preservation
Data publication
Local generalization
Local bucketization

ABSTRACT

Anonymization technique has been extensively studied and widely applied for privacy-preserving data publishing. In most previous approaches, a microdata table consists of three categories of attributes, namely explicit-identifier, quasi-identifier (QI), and sensitive attribute. In general, individuals may have different views on the sensitivity of different attributes. Therefore, there is another type of attribute that contains both QI values and sensitive values, termed semi-sensitive attribute. In this paper, we propose a new anonymization technique, called Local Generalization and Bucketization, to prevent identity disclosure and protect the sensitive values on each semi-sensitive attribute and sensitive attribute. The rationale is to use local generalization and local bucketization to divide the tuples into local equivalence groups and partition the sensitive values into local buckets, respectively. The protections of local generalization and local bucketization are independent, so that they can be implemented by appropriate algorithms without weakening other protection. Besides, the protection of local bucketization for each semi-sensitive attribute and sensitive attribute is also independent. Consequently, local bucketization can comply with various principles in different attributes according to the actual requirements of anonymization. We conducted extensive experiments to illustrate the effectiveness of the proposed approach.

1. Introduction

With human society entering the age of big data, a variety of individual information, such as income investigation, medical information, and population census, has been collected by corporations and governments. This massive personal data, also known as microdata, is used for data mining and machine learning that contributes to corporations creating business values and governments developing policies. However, the microdata always contains private information, resulting in the secrets of individuals being likely to be disclosed if the microdata is published without any disguise (Raju et al., 2022; Safi et al., 2022).

Many anonymization techniques, such as generalization (Sweeney, 2002) and bucketization (Xiao and Tao, 2006), are proposed for privacy-preserving data publishing. In these approaches, the attributes in the microdata table are classified into three categories: (1) Explicit-Identifier, which can uniquely or mostly identify the record owner and must be removed from the published table; (2) Quasi-Identifier (QI), which can be used to re-identify the record owner when taken together; and (3) Sensitive attribute, which contains the confidential information of individuals.

Generalization transforms the values on QI attributes into general forms, and the tuples whose generalized values are the same constitute an equivalence group. As a result, the records in the same equivalence group are indistinguishable. While bucketization divides the tuples into buckets that break the relation between QI attributes and sensitive attributes. Therefore, every record corresponds to the diverse sensitive values within the bucket.

1.1. Motivation

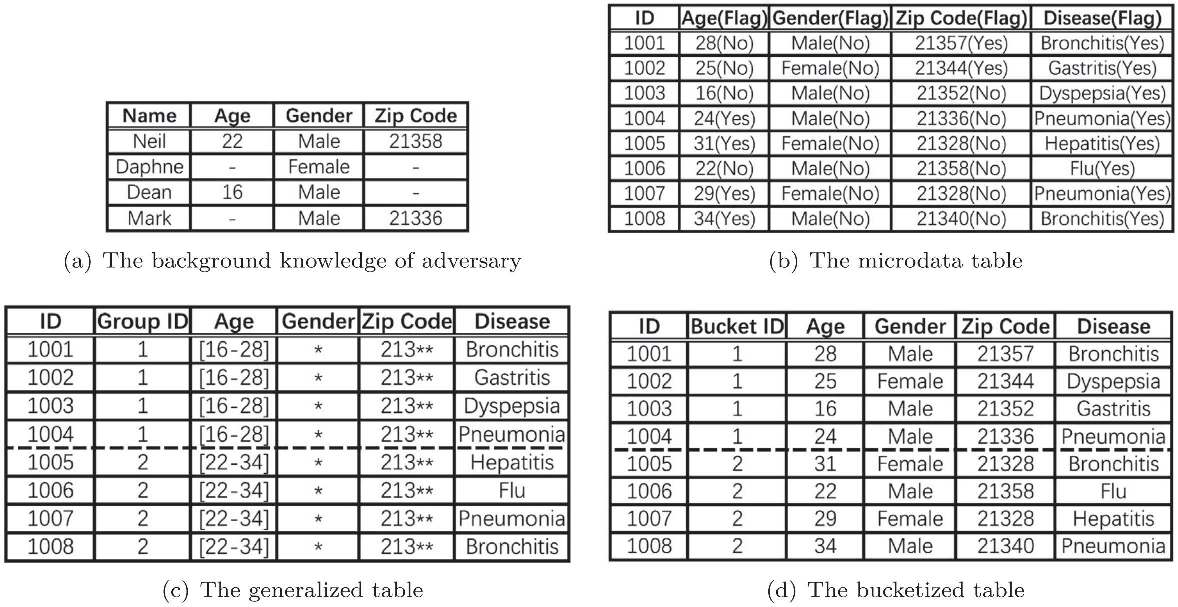
Previous approaches always suppose that an attribute includes only QI values or sensitive values. In fact, different individuals may view different data values as sensitive on the same attribute. Thus, an attribute may contain both QI values and sensitive values, which are considered as semi-sensitive. For example, a hospital releases some diagnosis records of patients, as shown in Fig. 1(b),

[*] Corresponding author.
E-mail address: brooklet60@hust.edu.cn (K. He).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2022.12.008



Name	Age	Gender	Zip Code
Neil	22	Male	21358
Daphne	-	Female	-
Dean	16	Male	-
Mark	-	Male	21336

(a) The background knowledge of adversary

ID	Age(Flag)	Gender(Flag)	Zip Code(Flag)	Disease(Flag)
1001	28(No)	Male(No)	21357(Yes)	Bronchitis(Yes)
1002	25(No)	Female(No)	21344(Yes)	Gastritis(Yes)
1003	16(No)	Male(No)	21352(No)	Dyspepsia(Yes)
1004	24(Yes)	Male(No)	21336(No)	Pneumonia(Yes)
1005	31(Yes)	Female(No)	21328(No)	Hepatitis(Yes)
1006	22(No)	Male(No)	21358(No)	Flu(Yes)
1007	29(Yes)	Female(No)	21328(No)	Pneumonia(Yes)
1008	34(Yes)	Male(No)	21340(No)	Bronchitis(Yes)

(b) The microdata table

ID	Group ID	Age	Gender	Zip Code	Disease
1001	1	[16-28]	*	213**	Bronchitis
1002	1	[16-28]	*	213**	Gastritis
1003	1	[16-28]	*	213**	Dyspepsia
1004	1	[16-28]	*	213**	Pneumonia
1005	2	[22-34]	*	213**	Hepatitis
1006	2	[22-34]	*	213**	Flu
1007	2	[22-34]	*	213**	Pneumonia
1008	2	[22-34]	*	213**	Bronchitis

(c) The generalized table

ID	Bucket ID	Age	Gender	Zip Code	Disease
1001	1	28	Male	21357	Bronchitis
1002	1	25	Female	21344	Dyspepsia
1003	1	16	Male	21352	Gastritis
1004	1	24	Male	21336	Pneumonia
1005	2	31	Female	21328	Bronchitis
1006	2	22	Male	21358	Flu
1007	2	29	Female	21328	Hepatitis
1008	2	34	Male	21340	Pneumonia

(d) The bucketized table

Fig. 1. An example of anonymizing microdata table.

to allow researchers to study the characteristics of various diseases. In the microdata table, each attribute except ID has a flag that marks whether a tuple treats her/his value as sensitive (e.g., the tuple with ID 1002 does not care if her age value is known by others, but the one with ID 1007 wants to keep secret). In consequence, the attributes of age and zip code are semi-sensitive in the microdata table because they contain both QI values and sensitive values.

Suppose that an adversary has the background knowledge as shown in Fig. 1(b), and obtains the microdata table of Fig. 1(b). Knowing that Mark went to the hospital before and matching by his values of gender and zip code, the adversary infers that: (1) his record is with ID 1004 in the microdata table; and (2) his age is 24, and the disease is pneumonia. The goal of preventing such privacy disclosures has resulted in the development of many anonymization techniques (see survey (Fung et al., 2010)). Previous generalization or bucketization anonymize whole attributes rather than specific values, then they can only regard semi-sensitive attributes as QI attributes. The generalized and bucketized versions of Fig. 1(b) are given in Fig. 1(c) and 1(d), respectively.

Although generalization effectively prevents identity disclosure, it always suffers from serious information loss, as proposed by Aggarwal (2005), Li and Li (2009), Kifer and Gehrke (2006). Almost all the values are irreversibly generalized, which hinders recipients from analyzing data information. For example, Fig. 1(c)(c) stops the adversary from recognizing the record owner but in which poor information utility is preserved for recipients.

While the bucketized table preserves excellent information utility, but it only protects the confidential values on the sensitive attribute without caring about personalized privacy requirements. The sensitive values on the semi-sensitive attributes tend to be revealed when the adversary has enough background knowledge. For example, the adversary can still acquire the ID and age of Mark by matching QI values in Fig. 1(d).

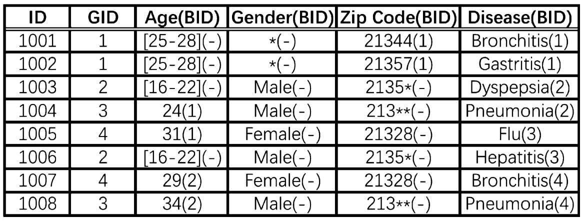
In this paper, we propose a new algorithm called **L**ocal **G**eneralization and **B**ucketization (LGB) to address the problem of protecting personalized privacy information. LGB prevents both disclosures of identities and sensitive values and also preserves significant information utility. It uses local generalization and local bucketization to partition the tuples into local equivalence groups in which just specific QI values are generalized and divide the sensitive values into local buckets within each semi-sensitive attribute and sensitive attribute, respectively. The detailed formalization and analysis of LGB are presented in Section 2. Fig. 2 shows a possible anonymized result of Fig. 1(b) by LGB.

In Fig. 2, the attribute of GID denotes the ID of local equivalence group, and the flag of BID represents the ID of local bucket inside the corresponding attribute. Note that, every local bucket contains only sensitive values, and all the QI values are generalized by local generalization. For example, when the adversary matches the QI values of Mark in Fig. 2, he can only infer that Mark's record ID may be 1004 and 1008, which belongs to the local equivalence group of GID 3, and the attribute of age includes the local buckets of BID 1 and 2 in the local equivalence group. Then the adversary concludes that Mark's age may be 24, 31, 29, and 34. For the same reason, the disease value of Mark may be dyspepsia, pneumonia, and bronchitis. As a result, the adversary can not determine the exact record ID and sensitive values of the target tuple.

1.2. Contributions

This study extends the concept of personalized anonymity (Xiao and Tao, 2006; Wang et al., 2009). It assumes that individuals can determine their sensitive values at will, an attribute can be QI, semi-sensitive, or sensitive, and a microdata table consists of several QI attributes, semi-sensitive attributes, and sensitive attributes. We suppose the background knowledge of adversary is that: (1) the adversary does not acquire any sensitive value on semi-sensitive attributes and sensitive attributes because people must cautiously keep their confidential information from strangers; and (2) in the worst case, the adversary knows the existences and QI values of all the individuals in microdata table. The adversary aims to obtain the ID and sensitive values of target person from the anonymized table. Our contributions are as follows.

First, we propose the LGB technique to protect personalized privacy information. LGB combines with local generalization and local bucketization to provide secure protections for identities and sensitive values, and it also reduces information loss as far as possible. The protections of local generalization and local bucketization are independent, so that they can be achieved by appropriate algorithms without weakening the other protection separately. Additionally, the protection of local bucketization in each



ID	GID	Age(BID)	Gender(BID)	Zip Code(BID)	Disease(BID)
1001	1	[25-28](-)	*(-)	21344(1)	Bronchitis(1)
1002	1	[25-28](-)	*(-)	21357(1)	Gastritis(1)
1003	2	[16-22](-)	Male(-)	2135*(-)	Dyspepsia(2)
1004	3	24(1)	Male(-)	213**(-)	Pneumonia(2)
1005	4	31(1)	Female(-)	21328(-)	Flu(3)
1006	2	[16-22](-)	Male(-)	2135*(-)	Hepatitis(3)
1007	4	29(2)	Female(-)	21328(-)	Bronchitis(4)
1008	3	34(2)	Male(-)	213**(-)	Pneumonia(4)

Fig. 2. The anonymized table by LGB.

semi-sensitive attribute and sensitive attribute is also independent. Therefore, local bucketization can comply with different principles in the different attributes according to the practical demands of anonymization.

Second, we illustrate the effective protections of LGB for identities and sensitive values based on satisfying the principles of k-anonymity and l-diversity, respectively, i.e., for each tuple, the probabilities of the disclosures of identity and sensitive values are at most $1/k$ and $1/l$, respectively. Moreover, since the protections are independent, either degree of protection can be flexibly adjusted according to the actual requirements without reducing the other level of defense severally.

Third, an efficient algorithm is presented to achieve LGB complying with k-anonymity and l-diversity. The algorithm contains two main parts, namely local generalization and local bucketization, to partition the tuples into local equivalence groups and divide the sensitive values into local buckets, respectively. We also propose two different algorithms to implement local generalization based on multi-dimensional partition and minimizing normalized certainty penalty (NCP) for different utilization purposes separately. Furthermore, the range of each local bucket is minimized as far as possible to preserve more information utility.

Last but not least, we conduct a large number of experiments to illustrate the basic property of LGB and the different performances between the two proposed local generalization algorithms through the results of discernibility metric measurement, NCP, and aggregate query answering. The effect of the density of the sensitive values in semi-sensitive attributes is also studied.

The rest of this paper is organized as follows. Section 2 proposes the formalization and analysis of LGB. Section 3 presents an algorithm to achieve LGB. Section 4 shows the results and analysis of experiments. Section 5 describes the related studies. Section 6 concludes the paper and proposes the directions for future studies.

2. Definitions and analysis

2.1. Concepts

The formalization of LGB technique requires certain prior and novel concepts, and the important symbols are summarized as shown in Table 1. We first re-define the categories of attributes based on the property of data value as follows.

Definition 1 (*QI Attribute*). An attribute is considered as a QI attribute, denoted as A^{QI}, if and only if the attribute contains only QI values.

Definition 2 (*Sensitive Attribute*). An attribute is considered as a sensitive attribute, denoted as A^{SA}, if and only if the attribute contains only sensitive values.

Definition 3 (*Semi-Sensitive Attribute*). An attribute is considered as a semi-sensitive attribute, denoted as A^{SS}, if and only if the attribute contains both QI values and sensitive values.

Table 1
Summary of notations.

Notation	Description
T	A microdata table
A^{QI}	A QI attribute
A^{SA}	A sensitive attribute
A^{SS}	A semi-sensitive attribute
G	A QI group
LEG	A local equivalence group
$t[A]$	The value of tuple t on attribute A
k	The parameter for controlling k-anonymity
l	The parameter for controlling l-diversity

The new definitions of attributes allow individuals to customize their own privacy requirements, and releaser can employ appropriate anonymization approaches to protect people's personalized privacy information and preserve worthy data utility according to the characters of different attributes. We define the personalized privacy preservation as follows.

Definition 4 (*Personalized Privacy Preservation*). Given a microdata table T consisting of several QI attributes, sensitive attributes and semi-sensitive attributes, personalized privacy preservation aims to prevent the identity disclosure and protect all the sensitive values for each tuple in T.

Next, we introduce the definition of QI Group as follows.

Definition 5 (*Partition and QI Group*). A partition consists of several subsets of T, such that each tuple belongs to exactly one subset, and each subset is called a QI group. Specifically, let there be m QI groups $\{G_1, G_2, \cdots, G_m\}$, then $\bigcup_{i=1}^{m} G_i = T$, and for any $1 \leqslant i_1 \neq i_2 \leqslant m, G_{i_1} \cap G_{i_2} = \varnothing$.

QI group has different performances when using different anonymization approaches. In generalization, the tuples in the same QI group have the same QI values. While in bucketization, each QI group is divided into two sub-tables, each containing QI values and sensitive values, respectively.

Definition 6 (*Equivalence Group*). Given a partition of T with m QI groups, each QI group is called an equivalence group, if for any tuple $t \in T$, a generalized table of T contains the tuple t of the form:

$$(G_j[1], G_j[2], \cdots, G_j[d], t[A^{SA}]),$$

where $G_j (1 \leqslant j \leqslant m)$ is the unique QI group including $t, G_j[i] (1 \leqslant i \leqslant d)$ is the generalized value on A_i^{QI} for all the tuples in G_j, and $t[A^{SA}]$ represents t's value on A^{SA}.

Definition 7 (*Bucket*). Given a partition of T with m QI groups, each QI group is called a bucket, if each QI group is represented as the form:

$$QIT(QI, BID) \text{ and } SAT(SA, BID),$$

where QI and SA are the QI values and sensitive values of the tuples in the QI group, respectively, and BID denotes the ID of bucket.

In previous equivalence groups, all the whole attributes including QI values are generalized to the same form, which always causes overprotection. We propose local generalization technique based on the new definitions of attributes to partition the tuples into local equivalence groups by generalizing just specific QI values.

Definition 8 (*QI Partition*). For any tuple $t \in T$, $QI[t]$ represents the set of the attributes containing t's QI values, such that

$$QI[t] = \{A|t[A] \text{ is a QI value}\}.$$

A QI partition of T divides the table into disjoint subsets $\{T_1, T_2, \cdots, T_m\}$, such that for any $1 \leqslant i \neq j \leqslant m, T_i \cap T_j = \emptyset$, $\bigcup_{i=1}^{m} T_i = T$, and for any $t_{i_1}, t_{i_2} \in T_i, QI[t_{i_1}] = QI[t_{i_2}]$.

Definition 9 (*Local Equivalence Group*). Given a microdata table T and a QI partition of T with m subsets, each subset is called a local equivalence group, if the QI values are generalized to the same form in the corresponding attribute, such that for any tuple $t \in T$, a locally generalized table of T contains the tuple t of the form:

$$(LEG_j[A_1^{QI}], \cdots, LEG_j[A_p^{QI}], t[A_1^{SA}], \cdots, t[A_q^{SA}]),$$

where $LEG_j(1 \leqslant j \leqslant m)$ is the unique local equivalence group including $t, A_{i_1}^{QI}(1 \leqslant i_1 \leqslant p)$ and $A_{i_2}^{SA}(1 \leqslant i_2 \leqslant q)$ denote the attributes containing QI value and sensitive value of t in LEG_j, respectively, $LEG_j[A^{QI}]$ is the generalized value on A^{QI} for all the tuples in LEG_j, and $t[A^{SA}]$ represents t's value on attribute A^{SA}.

Local generalization contains two partition steps. First, it divides the tuples into subsets by QI partition, in which all the records carry the QI values on the same attributes. For example, in Fig. 3(a), the tuples with ID 1001 and 1002 are in the same subset because both of them just carry QI values on the attributes of age and gender. Next, local generalization partitions the tuples into local equivalence groups within each subset and generalizes their QI values. For example, the tuples in the subset of GID 3 in Fig. 3 (a) are divided into the local equivalence groups of GID 3 and 4 in Fig. 3(b), and every group in Fig. 3(b) is a local equivalence group.

Likewise, the previous bucketization protects for the whole sensitives attribute rather than specific sensitive values. We present local bucketization technique to partition the sensitive values into local buckets in the corresponding attribute.

Definition 10 (*Local Bucket*). For any semi-sensitive attribute or sensitive attribute in T, the sensitive values are partitioned into local buckets, and each local bucket has the form:

$$IDT(ID, BID) \text{ and } SAT(SA, BID),$$

where ID and SA represent the IDs and sensitive values of the tuples in the local bucket, respectively, and BID denotes the ID of local bucket within the attribute.

For example, in Fig. 4, the tuples with ID 1004 and 1005 are in the same local bucket of BID 1 within the attribute of age, but they are in the different local buckets of BID 2 and 3 within the attribute of disease. Therefore, the local buckets within the different attributes are independent.

Note that, the previous equivalence groups and buckets can be treated as the special cases of local equivalence groups and local buckets when the microdata table does not contain any semi-sensitive attribute, respectively. Based on Definition 9 and 10, we define LGB technique as follows.

Definition 11 (*Local Generalization and Bucketization*). Given a microdata table T, a local generalization and bucketization of T is given by the partitions of local generalization and local bucketization, and every tuple and sensitive value belongs to exactly one local equivalence group and local bucket, respectively.

2.2. Protection analysis

In this section, we analyze the protections of local generalization and local bucketization against the disclosures of identities and sensitive values in detail. Without loss of generality, we illustrate how local generalization and local bucketization comply with k-anonymity and l-diversity[1], respectively. The definitions of k-anonymity and l-diversity are introduced as follows.

Definition 12 (*k-Anonymity*). A microdata table T satisfies k-anonymity, if for any tuple $t \in T$, the probability of identity disclosure is less than or equal to $1/k$.

Definition 13 (*l-Diversity*). A microdata table T complies with l-diversity, if the probability that any sensitive value is disclosed is less than or equal to $1/l$.

Then, we prove the locally generalized and bucketized table can also satisfy k-anonymity and l-diversity by meeting the corresponding conditions. We first consider the protection against identity disclosure in the locally generalized table and have the following lemma and corollary.

Lemma 1. *Given a locally generalized table, for any tuple $t \in T$, the probability of identity disclosure is at most $1/|LEG(t)|$, where $LEG(t)$ is the local equivalence group including t.*

Proof. According to Definition 9, the tuples in the same local equivalence group have the same attributes containing QI values, and all the QI values are generalized to the same form. Consequently, the adversary must obtain at least $|LEG(t)|$ possible tuples by matching QI values, then the probability of identity disclosure is at most $1/|LEG(t)|$. □

Corollary 1. A locally generalized table complies with k-anonymity, if every local equivalence group contains at least k tuples.

Proof. Given a locally generalized table in which every local equivalence group includes at least k tuples, for any tuple $t \in T$, we have

$$|LEG(t)| \geqslant k,$$

where $|LEG(t)|$ is the size of the local equivalence group including t. Then

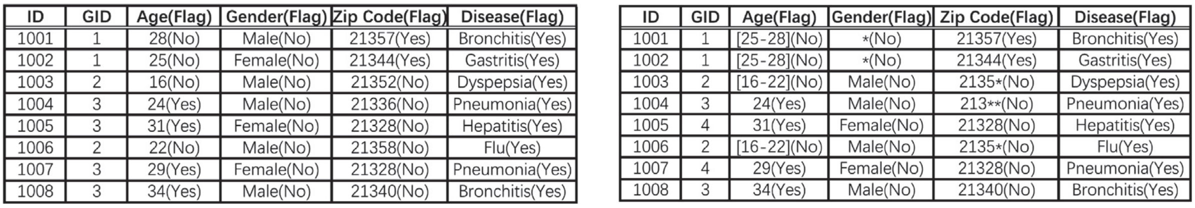
$$\frac{1}{|LEG(t)|} \leqslant \frac{1}{k}.$$

According to Lemma 1, the probability of identity disclosure for any tuple is at most $1/k$. Therefore, the locally generalized table complies with k-anonymity. □

Next, we discuss the protection for sensitive values in the locally bucketized table. Suppose an adversary knows t's existence and QI values, then attempts to infer t's sensitive value s from the locally bucketized table. The adversary needs to find t's possible records in the locally bucketized table by matching the QI values.

Definition 14 (*Matching Tuple*). Given a locally bucketized table T^{buc}, and for any tuple $t \in T$, a tuple $mt \in T^{buc}$ is a matching tuple of t if each QI value of t matches that of mt.

[1] In this paper, we use frequency l-diversity to confine LGB, and other versions of l-diversity (e.g., entropy l-diversity) can also be applied.



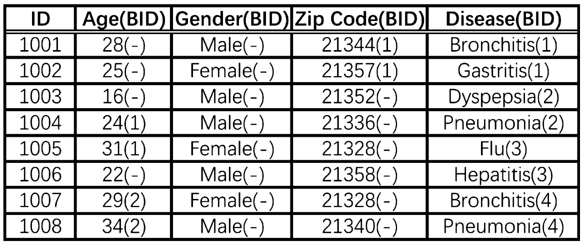
ID	GID	Age(Flag)	Gender(Flag)	Zip Code(Flag)	Disease(Flag)
1001	1	28(No)	Male(No)	21357(Yes)	Bronchitis(Yes)
1002	1	25(No)	Female(No)	21344(Yes)	Gastritis(Yes)
1003	2	16(No)	Male(No)	21352(No)	Dyspepsia(Yes)
1004	3	24(Yes)	Male(No)	21336(No)	Pneumonia(Yes)
1005	3	31(Yes)	Female(No)	21328(No)	Hepatitis(Yes)
1006	2	22(No)	Male(No)	21358(No)	Flu(Yes)
1007	3	29(Yes)	Female(No)	21328(No)	Pneumonia(Yes)
1008	3	34(Yes)	Male(No)	21340(No)	Bronchitis(Yes)

(a) The first partition

ID	GID	Age(Flag)	Gender(Flag)	Zip Code(Flag)	Disease(Flag)
1001	1	[25-28](No)	*(No)	21357(Yes)	Bronchitis(Yes)
1002	1	[25-28](No)	*(No)	21344(Yes)	Gastritis(Yes)
1003	2	[16-22](No)	Male(No)	2135*(No)	Dyspepsia(Yes)
1004	3	24(Yes)	Male(No)	213**(No)	Pneumonia(Yes)
1005	4	31(Yes)	Female(No)	21328(No)	Hepatitis(Yes)
1006	2	[16-22](No)	Male(No)	2135*(No)	Flu(Yes)
1007	4	29(Yes)	Female(No)	21328(No)	Pneumonia(Yes)
1008	3	34(Yes)	Male(No)	21340(No)	Bronchitis(Yes)

(b) The second partition

Fig. 3. An example of local generalization.



ID	Age(BID)	Gender(BID)	Zip Code(BID)	Disease(BID)
1001	28(-)	Male(-)	21344(1)	Bronchitis(1)
1002	25(-)	Female(-)	21357(1)	Gastritis(1)
1003	16(-)	Male(-)	21352(-)	Dyspepsia(2)
1004	24(1)	Male(-)	21336(-)	Pneumonia(2)
1005	31(1)	Female(-)	21328(-)	Flu(3)
1006	22(-)	Male(-)	21358(-)	Hepatitis(3)
1007	29(2)	Female(-)	21328(-)	Bronchitis(4)
1008	34(2)	Male(-)	21340(-)	Pneumonia(4)

Fig. 4. The locally bucketized table.

Definition 15 (*Matching Bucket*). Given a locally bucketized table T^{buc}, and for any tuple $t \in T$, a local bucket mb within an attribute is a matching bucket of t if there is at least a matching tuple of t in mb.

For example, when the adversary matches Mark in the locally bucketized table of Fig. 4, he can infer that Mark's matching tuple is with ID 1004, and the matching buckets inside the attributes of age and disease are with ID 1 and 2, respectively.

We denote $p(t,s)$ as the probability that the sensitive value s of t is exposed, and let $p(t,b)$ represent the probability that t is in the bucket b. Then we have the following lemma and corollary.

Lemma 2. *Given a locally bucketized table, for any tuple $t \in T$, the probability that any sensitive value s of t is exposed is as follows:*

$$p(t,s) \leqslant \sum_{mb} p(t,mb)\frac{|mb(s')|}{|mb|},$$

where $|mb(s')|$ is the number of the most occurrence sensitive value s' in the matching bucket mb, and $|mb|$ is the size of mb.

Proof. To acquire the sensitive value s, the adversary has to calculate the probabilities that t exists in each local bucket and t carries the sensitive value s within each local bucket. Then, the adversary has:

$$p(t,s) = \sum_{B} p(t,b)p(s|t,b),$$

where $p(s|t,b)$ denotes the probability that t carries the sensitive value s given that t is in the local bucket b. The adversary eliminates the local bucket that does not contain any matching tuple of t, expressed as follows:

$$p(t,b) = 0,\ \text{if}\ \neq xistsmt \in b.$$

According to Definition 15, we have:

$$p(t,s) = \sum_{mb} p(t,mb)p(s|t,mb),$$

The most occurrence sensitive value s' in mb is expressed as:

$$|mb(s)| \leqslant |mb(s')|.$$

Thus:

$$p(s|t,mb) = \frac{|mb(s)|}{|mb|} \leqslant \frac{|mb(s')|}{|mb|},$$

then:

$$p(t,s) = \sum_{mb} p(t,mb)p(s|t,mb) \leqslant \sum_{mb} p(t,mb)\frac{|mb(s')|}{|mb|}.$$

□

Corollary 2. A locally bucketized table complies with l-diversity principle if every local bucket satisfies the conditions: (1) each sensitive value appears at most once in the local bucket; and (2) the size of each local bucket is at least l.

Proof. According to Lemma 2, for any tuple $t \in T$, we have $p(t,s) \leqslant \sum_{mb} p(t,mb)\frac{|mb(s')|}{|mb|}$.

We confine that each sensitive value appears at most once inside the local bucket, such that for any $s \in T$,

$$|mb(s)| \leqslant 1.$$

And for any local bucket b, we have:

$$p|b| \geqslant l.$$

Then:

$$p(t,s) \leqslant \sum_{mb} p(t,mb)\frac{|mb(s')|}{|mb|} \leqslant \frac{1}{l}\sum_{mb} p(t,mb) = \frac{1}{l}.$$

In consequence, the locally bucketized table complies with l-diversity by meeting the conditions. □

Finally, we prove that a locally generalized and bucketized table complies with k-anonymity and l-diversity by satisfying the conditions in Corollary 1 and 2.

Corollary 3. A locally generalized and bucketized table complies with k-anonymity and l-diversity by meeting the conditions as follows: (1) each local equivalence group contains at least k tuples; (2) each sensitive value appears at most once inside each local bucket; and (3) the size of each local bucket is at least l.

Proof. According to Definition 11, the protections of local generalization and local bucketization are independent, and the conditions in Corollary 1 and 2 are non-overlapping. As a result, as long as the locally generalized and bucketized table satisfies the corresponding conditions, it complies with k-anonymity and l-diversity. □

Generally, local generalization enhances the protection of local bucketization because local generalization transforms QI values into the same form that increases the number of the matching tuples of target tuple. For example, the tuple with ID 1006 is in the local bucket of BID 3 within the attribute of disease in Fig. 4, while he is in that of BID 2 and 3 in Fig. 2, because his QI values are generalized to the same form in the local equivalence group of GID 2 that increases the number of his matching tuples. Then the probability that his disease value is disclosed is decreased to 1/4.

3. Proposed method

This section presents an algorithm to achieve LGB complying with k-anonymity and l-diversity. In addition, two algorithms are proposed to implement local generalization for different utilization purposes. The main procedure of LGB is given in Algorithm 1.

Algorithm 1. LGB

```
Input: microdata table $T$, parameters $k$ and $l$
Output: anonymized table $T_{anony}$
1: $Attri_{sen} = \{the\ attributes\ including\ sensitive\ values\}$
2: $T_{anony} = T$
3: for each $attr \in Attri_{sen}$ do
4:     $ValuePair_{sen} = \{(id, s) | s \in attr\ and\ s\ is\ sensitive\}$
5:     $T_{anony} = local\_bucketization(T_{anony}, ValuePair_{sen}, l)$
6: end for
7: $T_{anony} = local\_generalization(T_{anony}, k)$
8: return $T_{anony}$
```

The data structure $Attri_{sen}$ (line 1) stores the attributes including sensitive values in T, i.e., the set of semi-sensitive attributes and sensitive attributes. The variable T_{anony} (line 2) denotes the anonymized result, and it is initialized as T. In each iteration (lines 3 to 6), the algorithm picks an attribute from $Attri_{sen}$ and chooses the tuples with sensitive values (line 4). Then the algorithm divides the tuples into local buckets based on the value of l (line 5). After the loop, the function *local_generalization* divides T_{anony} into local equivalence groups according to the value of k (line 7). Finally, the algorithm returns T_{anony} as the anonymized result of T (line 8). Note that, in Algorithm 1, the QI attributes are not contained in $Attri_{sen}$, and $ValuePair_{sen}$ does not include any tuple with QI value in *attr*. Therefore, none of the local buckets contains any QI value.

The procedure comprises two main parts, namely local bucketization (line 5) and local generalization (line 7). We elaborate each part in the rest of this section.

3.1. Local bucketization

This section presents an efficient algorithm to implement the function *local_bucketization* in Algorithm 1. To preserve more information utility, the algorithm partitions the tuples in $ValuePair_{sen}$ into local buckets and minimizes the range of the sensitive values in each local bucket as far as possible. The detailed procedure is shown in Algorithm 2.

Algorithm 2. local_bucketization

```
Input: anonymized table $T_{anony}$, pairs of values $ValuePair$,
   parameter $l$
Output: anonymized table $T_{anony}$
1: $value\_number = \{(value, number) | count\ in\ ValuePair\}$
2: $median = calculate\_median(value\_number)$
3: $VP_{small} = \{(id, s) | (id, s) \in ValuePair\ and\ s \leqslant median\}$
4: $VP_{big} = \{(id, s) | (id, s) \in ValuePair\ and\ s > median\}$
5: if $check(VP_{small}, l)$ and $check(VP_{big}, l)$ then
6:     $T_{anony} = local\_bucketization(T_{anony}, VP_{small}, l)$
7:     $T_{anony} = local\_bucketization(T_{anony}, VP_{big}, l)$
8: else
9:     $divide\_buckets(T_{anony}, ValuePair, l)$
10: end if
11: return $T_{anony}$
```

The algorithm recursively divides *ValuePair* into two smaller sets, and their ranges of sensitive values are non-overlapping. The data structure *value_number* stores every sensitive value and the number of occurrence counted in *ValuePair* (line 1). The algorithm calculates the median value of the sensitive values based on their numbers in *value_number* (line 2), and divides *ValuePair* into two smaller sets (lines 3 and 4). The function *check* examines whether *ValuePair* can be divided into local buckets complying with l-diversity (line 5), such that the product of the number of the most occurrence sensitive value and the value of l is not more than the size of *ValuePair*. If both VP_{small} and VP_{big} meet the condition, the algorithm makes recursive calls the function *local_bucketization* for VP_{small} and VP_{big} (lines 6 and 7). Otherwise, the algorithm divides the tuples in *ValuePair* into local buckets (line 9). The function *divide_buckets* is implemented by the assignment algorithm of m-Invariance (Xiao and Tao, 2007) to satisfy the conditions in Corollary 2.

Proposition 1. *T_{anony} complies with l-diversity after local bucketization phase.*

Proof. The function *divide_buckets* is implemented by the assignment algorithm of m-Invariance, where the parameters l and m are mathematically equivalent. In each recursion of Algorithm 2, both sets of VP_{small} and VP_{big} are checked to satisfy l-eligible condition (Xiao and Tao, 2007). According to m-Invariance, the assignment algorithm divides the tuples into m-unique buckets. Then the size of each generated local bucket is at least l, and every sensitive value appears at most once in each local bucket. As a result, all the local buckets within the corresponding attribute meet the conditions in Corollary 2 based on the value of l. After the loop (lines 3 to 6) in Algorithm 1, the tuples with sensitive values inside each semi-sensitive attribute and sensitive attribute are partitioned into l-unique local buckets, so that T_{anony} satisfies l-diversity principle. □

3.2. Local generalization

In this section, we propose two algorithms to implement local generalization based on multi-dimensional partition and minimizing NCP separately. The experiments we conducted in Section 4 will show their different performances. We elaborate each algorithm as follows.

3.2.1. Based on multi-dimensional partition

Previous multi-dimensional partition (LeFevre et al., 2006) is an effective and popular approach to divide the tuples into equivalence groups. However, it does not apply to the publishing scenario of personalized privacy requirements. We combine multi-dimensional partition and QI partition to divide the tuples into local equivalence groups according to their specific QI values.

Definition 16 (*Multi-Dimensional QI Partition*). Given a microdata table T with d attributes and a QI partition with m subsets $\{T_1, T_2, \cdots, T_m\}$, a multi-dimensional QI partition divides the tuples into non-overlapping multi-dimensional regions within each subset that covers $D[A_1^i] \times D[A_2^i] \times \cdots \times D[A_d^i](1 \leqslant i \leqslant m)$, where $D[A_j^i](1 \leqslant j \leqslant d)$ denotes the domain of attribute A_j inside subset T_i, and for any tuple $t \in T, (t[A_1], t[A_2], \cdots, t[A_d])$ is mapped in the unique region.

To reduce information loss as far as possible, the size of each local equivalence group should be minimized to only satisfy k-anonymity, so that each region is divided into smaller ones until at least one of their sizes is less than k. Algorithm 3 describes the local generalization based on multi-dimensional partition in detail.

Algorithm 3. local_generalization

```
Input: microdata table $T$, parameter $k$
Output: anonymized table $T_{anony}$
 1: $T_{anony} = \varnothing$
 2: $T_{subsets} = \{the\ subsets\ of\ T\ divided\ by\ QI\ partition\}$
 3: for each $T_{set} \in T_{subsets}$ do
 4:   $partition\_set = \{T_{set}\}$
 5:   while $partition\_set \neq \varnothing$ do
 6:     $oper\_set = pick\_set(partition\_set)$
 7:     $partition\_set = partition\_set - oper\_set$
 8:     $QI\_set = QI[oper\_set]$
 9:     $partition\_flag = false$
10:     while $QI\_set \neq \varnothing$ do
11:       $attri = choose\_dimension(oper\_set, QI\_set)$
12:       $split\_value = cal\_median(oper\_set, attri)$
13:       $S_l = \{t | t \in oper\_set\ and\ t[attri] \leqslant split\_value\}$
14:       $S_r = \{t | t \in oper\_set\ and\ t[attri] > split\_value\}$
15:       if $|S_l| \geqslant k$ and $|S_r| \geqslant k$ then
16:         $partition\_set = partition\_set + S_l$
17:         $partition\_set = partition\_set + S_r$
18:         $partition\_flag = true$
19:         break
20:       else
21:         $QI\_set = QI\_set - attri$
22:       end if
23:     end while
24:     if $partition\_flag$ is false then
25:       $gen\_set = generalize(oper\_set)$
26:       $T_{anony} = T_{anony} + gen\_set$
27:     end if
28:   end while
29: end for
30: return $T_{anony}$
```

The data structures T_{anony} and $T_{subsets}$ store the anonymized result and the subsets of T divided by QI partition, respectively (lines 1 and 2). In each iteration (lines 3 to 29), the algorithm picks and divides a subset T_{set} into local equivalence groups. The data structure *partition_set* contains the sets of tuples which has not been generalized, and it includes T_{set} in the beginning (line 4). As long as *partition_set* is not empty (line 5), the algorithm picks and eliminates a set from *partition_set* (lines 6 and 7). The data structure *QI_set* denotes the set of the attributes including QI values in *oper_set* (line 8), and *partition_flag* represents whether *oper_set* can be divided (line 9). While *QI_set* is not empty (line 10), the algorithm chooses an attribute *attri* and calculates the median value (lines 11 and 12), then divides *oper_set* into two smaller sets (lines 13 and 14). If the sizes of S_l and S_r are both bigger than or equal to k (line 15), the algorithm adds S_l and S_r into *partition_set* (lines 16 and 17) and sets *partition_flag* as *true* (line 18), then breaks the while loop (line 19). Otherwise, *attri* is eliminated from *QI_set* (line 21). If *partition_flag* is false after the while loop, not a single attribute can be used to divide *oper_set* into the smaller sets that complies with k-anonymity (line 24). The algorithm generalizes the QI values in *oper_set* and adds the generalized set *gen_set* into T_{anony} (lines 25 and 26). Finally, the algorithm returns T_{anony} as the generalized result (line 30).

Local generalization based on the multi-dimensional partition is well suited as a common approach because it evenly divides the tuples into local equivalence groups, which reduces much information loss. In practice, some microdata tables are published for particular purposes, and the information utility should be evaluated by a given metric. Next, we present another algorithm to achieve local generalization, which preserves information utility evaluated by a specific metric as far as possible.

3.2.2. Based on minimizing NCP

In this section, we propose a utility-based algorithm to implement local generalization by using NCP (Xu et al., 2006) as an information metric. For any tuple $t \in T$, t's value v is generalized to v^* on the categorical attribute A_{cat}, then.

$$NCP_{A_{cat}}(t) = \frac{size(v^*)}{|A_{cat}|},$$

where $size(v^*)$ is the number of the leaf nodes that are the descendants of v^* in the hierarchy tree of A_{cat}, and $|A_{cat}|$ denotes the number of distinct values on attribute A_{cat}. While t's value v is generalized to $[v^*_{lower}, v^*_{upper}]$ on the numeric attribute A_{num}, then

$NCP_{A_{num}}(t) = \frac{v^*_{upper} - v^*_{lower}}{range(A_{num})}$, where v^*_{lower} and v^*_{upper} are the lower bound and upper bound of generalized range, respectively, and $range(A_{num})$ is the range of all the values on attribute A_{num}. The information loss of whole generalized table is represented as

$$NCP(T) = \sum_{t \in T} \sum_{i=1}^{d} NCP_{A_i}(t).$$

The local generalization complying with k-anonymity based on minimizing NCP is described as Algorithm 4. The data structures T_{anony} and $T_{subsets}$ store the anonymized result and the subsets of T divided by QI partition, respectively (lines 1 and 2). In each iteration (lines 3 to 24), the algorithm picks and divides a subset T_{set} into local equivalence groups (line 3). The data structure *QI_set* denotes the set of the attributes including QI values in T_{set} (line 4), and *partition_set* represents the sets of tuples which has not been generalized (line 5). While *partition_set* is not empty (line 6), the algorithm picks and eliminates a set *oper_set* from *partition_set* (lines 7 and 8). If the size of *oper_set* is smaller than $2k$ (line 9), the tuples in *oper_set* are generalized and added into T_{anony} (lines 10 and 11). Otherwise, the function *find_seeds* returns two seed records that maximize the value of NCP based on *QI_set* (line 13), and the algorithm divides *oper_set* into two smaller sets according to the seed tuples (line 14). The functions *find_seeds* and *divide_table* can be implemented by the top-down algorithm in (Xu et al., 2006). Next, if the sizes of T_1 and T_2 are both more than or equal to k, the algorithm adds them into *partition_set* (lines 15 to 17), or else, *oper_set* is generalized and added into T_{anony} (lines 18 to 20). Finally, the algorithm returns T_{anony} as the generalized result (line 25).

Algorithm 4. local_generalization

```
Input: microdata table $T$, parameter $k$
Output: anonymized table $T_{anony}$
$T_{anony} = \varnothing$
$T_{subsets} = \{the\ subsets\ of\ T\ divided\ by\ QI\ partition\}$
for each $T_{set} \in T_{subsets}$ do
    $QI\_set = QI[T_{set}]$
    $partition\_set = \{T_{set}\}$
    while $partition\_set \neq \varnothing$ do
        $oper\_set = pick\_set(partition\_set)$
        $partition\_set = partition\_set - oper\_set$
        if $|oper\_set| < 2k$ then
            $gen\_set = generalize(oper\_set)$
            $T_{anony} = T_{anony} + gen\_set$
        else
            $sd_1, sd_2 = find\_seeds(oper\_set, QI\_set)$
            $T_1, T_2 = divide\_table(oper\_set, sd_1, sd_2, QI\_set)$
            if $|T_1| \geqslant k$ and $|T_2| \geqslant k$ then
                $partition\_set = partition\_set + T_1$
                $partition\_set = partition\_set + T_2$
            else
                $gen\_set = generalize(oper\_set)$
                $T_{anony} = T_{anony} + gen\_set$
            end if
        end if
    endwhile
end for
return $T_{anony}$
```

Proposition 2. *T_{anony} complies with k-anonymity after local generalization phase.*

Proof. In both Algorithm 3 and 4, the size of each subset must be more than or equal to k prior to putting it into *partition_set*, and only *oper_set* selected from *partition_set* is generalized and added into T_{anony}. Therefore, the sizes of all the local equivalence groups are at least k that satisfies the condition in Corollary 1. Then T_{anony} satisfies k-anonymity principle. □

4. Experiments

This section evaluates the efficiency of LGB technique. We use the real US Census data (Ruggles et al., 2020), eliminate the tuples with missing values, and randomly select 31,055 tuples with nine attributes. The QI attributes are relationship, marital status, race, education, and hours per week, the semi-sensitive attributes are sex, age, and occupation, and the sensitive attribute is salary. Table 2 describes the attributes in detail.

In Section 3, we propose an algorithm to implement LGB that satisfies k-anonymity to prevent identity disclosure and l-diversity to protect sensitive values. However, the attribute of sex contains only two distinct sensitive values, namely, male and female, so that complying with l-diversity principle can not provide effective protection. We severally employ t-closeness measured by EMD (Li et al., 2007) to confine the attribute of sex and l-diversity to protect the rest semi-sensitive attributes and sensitive attribute. In addition, the algorithms based on multi-dimensional partition and minimizing NCP are denoted as LGB_MDP and LGB_NCP, respectively. The experiments will show: (1) the basic property of LGB technique and the different performances between LGB_MDP and LGB_NCP; and (2) the effect of the density of sensitive values in the semi-sensitive attributes.

4.1. Information metrics

In this section, the experiments employ two information metrics, namely discernibility metric measurement (Bayardo and Agrawal, 2005) and NCP, to check the information utility. The parameters k is assigned to 5, 8, and 10, and l is assigned to 5, 8, 10, 12, 15, 18, and 20, respectively, and t is fixed at 0.2. Besides, the percentage of the sensitive values in each semi-sensitive attribute is set to about 20%.

The first experiment uses discernability metric measurement, denoted as C_{DM}, which is given by the equation:

$$C_{DM} = \sum_{LEG} |LEG|^2,$$

where *LEG* is the local equivalence group, and $|LEG|$ denotes the size of *LEG*. A smaller value of C_{DM} means less generalization and perturbation in the anonymization process. Fig. 5(a) and 5(b) present the results of the anonymized table by LGB_MDP and LGB_NCP, respectively.

It is obvious that the discernability penalty of LGB_MDP is much lower than that of LGB_NCP, so that LGB_MDP divides the tuples into local equivalence groups more evenly than LGB_NCP. Besides, the sizes of local equivalence groups are hardly affected by l because the values of discernability penalty are the same when the value of k is fixed. Therefore, the results indicate that the protection of local generalization is barely affected by local bucketization.

Next, the experiment uses NCP as an information metric to compare the information quality between LGB_MDP and LGB_NCP. Fig. 6(a) and 6(b) present the results of LGB_MDP and LGB_NCP, respectively. It is shown that the values of LGB_NCP are clearly lower than that of LGB_MDP by about 14% to 17% although LGB_MDP divides the tuples into local equivalence groups more evenly than LGB_NCP. Thus, the utility-based heuristic of LGB_NCP plays a role in reducing NCP. It proves that LGB is a flexible framework that can be implemented by appropriate algorithms to meet different actual demands.

4.2. Query answering

In this experiment, we use the approach of aggregate query answering (Zhang et al., 2007) to check information utility. We randomly generate 1,000 queries and calculate the average relative error for each anonymized table. The sequence of a query is expressed as the form:

SELECT SUM(salary) FROM **Microdata**
WHERE $pred(A_1)$ AND $pred(A_2)$ AND $pred(A_3)$ AND $pred(A_4)$.

Specifically, the query condition contains four random QI attributes or semi-sensitive attributes, and the sum of salary is the result for comparison. For categorical attributes, the predicate $pred(A)$ has the following form:

$$(A = v_1 \text{ or } A = v_2 \text{ or } \cdots \text{ or } A = v_m),$$

where $v_i (1 \leqslant i \leqslant m)$ is a random value from $D[A]$. While for numerical attributes, the predicate $pred(A)$ has the following form:

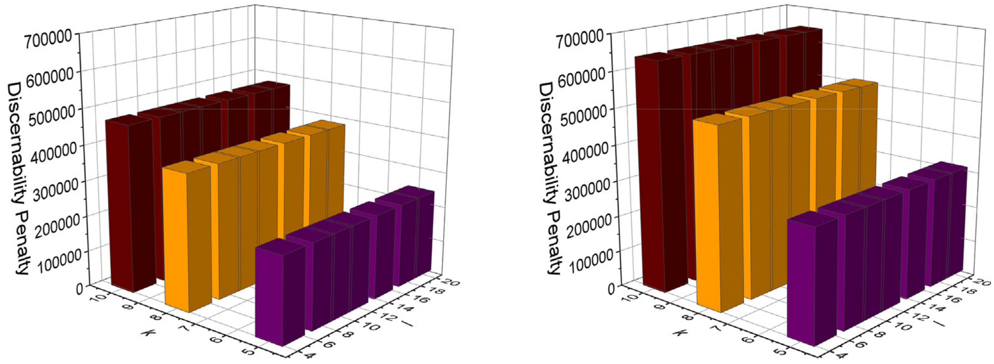
$$(A > v) \text{ or } (A < v) \text{ or } (A = v)$$

$$\text{or } (A \geqslant v) \text{ or } (A \leqslant v) \text{ or } (A \neq v),$$

where v is a random value from $D[A]$. According to (Zhang et al., 2007), the relative error rate, denoted as R_{error}, is given by the equation:

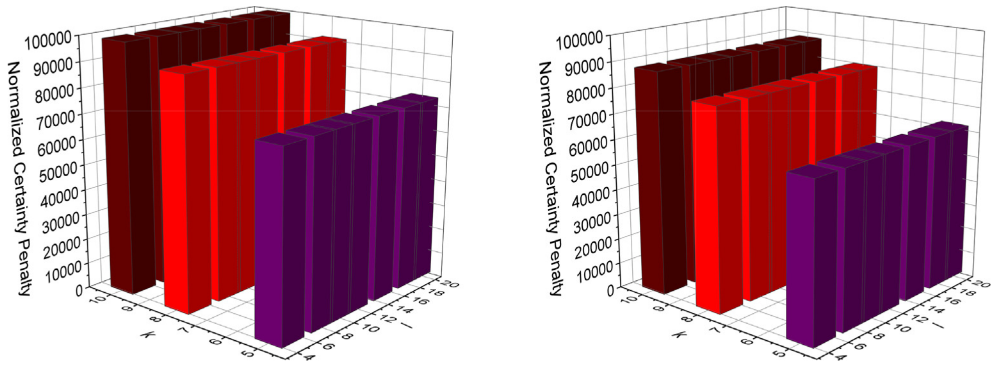
Table 2
Description of the attributes.

	Attribute	Value Type	Sensitivity Type	Size
1	Sex	Categorical	Semi-sensitive	2
2	Age	Continuous	Semi-sensitive	73
3	Relationship	Categorical	QI	13
4	Marital status	Categorical	QI	6
5	Race	Categorical	QI	9
6	Education	Categorical	QI	11
7	Hours per week	Continuous	QI	93
8	Occupation	Categorical	Semi-sensitive	257
9	Salary	Continuous	Sensitive	719



(a) Results of LGB_MDP (b) Results of LGB_NCP

Fig. 5. Discernibility metric results.



(a) Results of LGB_MDP (b) Results of LGB_NCP

Fig. 6. NCP results.

$$R_{error} = (Sum_{upper} - Sum_{lower})/Sum_{act},$$

where Sum_{upper} and Sum_{lower} are the upper bound and lower bound of the sum of salary, respectively, and Sum_{act} is the actual value.

According to (Zhang et al., 2007), Sum_{upper} and Sum_{lower} are calculated through "numbers of hits" in the group. However, since each tuple is contained in the local buckets of different semi-sensitive attributes, we need to count the upper bound and lower bound of the possible numbers of the tuples that satisfy the query conditions in each local bucket of salary. Let $Pro_t(A)$ denote the probability that t matches the condition on attribute A, and $Pro(t)$ represents the probability that t satisfies the query conditions, then we have.

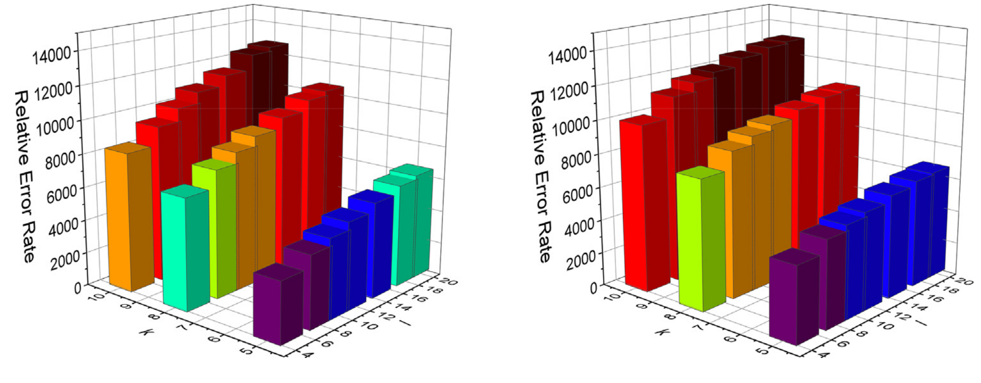
$$Pro(t) = \prod_{A} Pro_t(A).$$

For each local bucket within salary, we calculate the sum of probabilities, then round down and round up the sum as the lower bound and upper bound of the number of tuples, respectively. Next, we count the lower bound and upper bound of the sum of salary in the local bucket through the help table, denoted as $Sum_{lower}(b)$ and $Sum_{upper}(b)$, respectively. The lower bound and upper bound of the sum of salary in the whole table are expressed as.

$$Sum_{lower} = \sum_{B} Sum_{lower}(b),$$

and

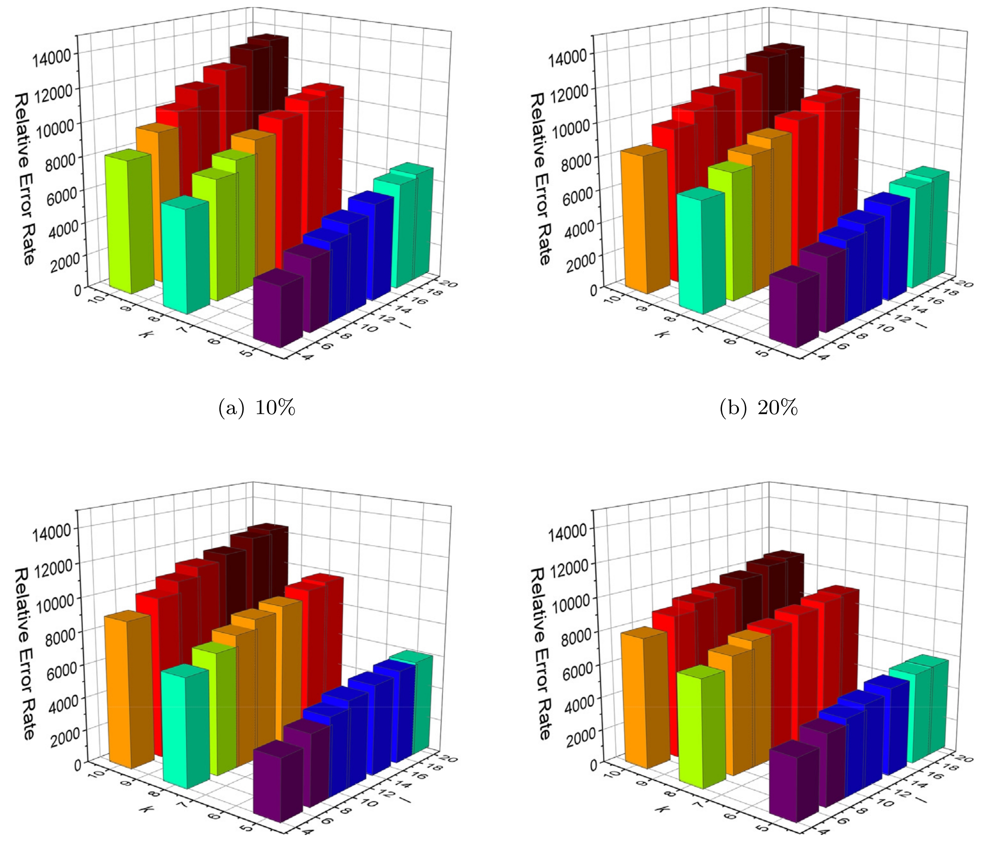
$$Sum_{upper} = \sum_{B} Sum_{upper}(b).$$



(a) Results of LGB_MDP

(b) Results of LGB_NCP

Fig. 7. Query answering results.



(a) 10%

(b) 20%

(c) 30%

(d) 40%

Fig. 8. Different density of sensitive values.

Fig. 7(a) and 7(b) show the results of aggregate query answering of LGB_MDP and LGB_NCP, respectively. The relative error rate increases with the growth of the value of k or l, and it is more affected by k than that of l. Consequently, local generalization has more influence on information utility than local bucketization. Note that, LGB_MDP has lower relative error rates than LGB_NCP when l is small because LGB_MDP divides the tuples into local equivalence groups more evenly than LGB_NCP that narrows the ranges of generalized values. But with the growth of l, the relative error rates of LGB_MDP are gradually close to that of LGB_NCP, because the ranges of local buckets are large enough to cover the query conditions, then local bucketization has more and more important influence on information utility. In conclusion, LGB_MDP is better than LGB_NCP if the value of l is small. Otherwise, LGB_MDP and LGB_NCP perform almost equally.

4.3. Effect of density

We check the effect of the density of the sensitive values in semi-sensitive attributes on LGB_MDP. The percentages of sensitive values are set to about 10%, 20%, 30%, and 40% in each semi-sensitive attribute, respectively, and the rest configurations are the same as the previous experiments. The results are evaluated through aggregate query answering, and Fig. 8(a), 8(b), 8(c), and

8(d) present the results of the percentages of 10%, 20%, 30%, and 40%, respectively.

With the density of sensitive values increases, the relative error rate is declined. The most obvious reduction is in Fig. 8(d), where the percentage of sensitive values is about 40%, and the values of k and l are 10 and 20, respectively. It is mainly because more values are divided into local buckets rather than that of equivalence groups, and the values are indistinguishable in equivalence groups but are specific in local buckets, so that less number of QI values narrow the ranges of generalized values, and more sensitive values do not lose the accuracies in local buckets and even compose more local buckets in which the ranges are further decreased.

5. Related work

Privacy-preserving data publishing determines an optimal trade-off between privacy protection and information preservation. Above all, the most important factor in applying appropriate anonymization technique is assuming the publishing scenario. Most researches focus on anonymizing a single and static release so far. However, there are more complicated data publishing environments in actual applications, such as multiple release publishing (Wang and Fung, 2006), continuous data publishing (Byun et al., 2006; Le et al., 2018), and collaborative data publishing (Wang et al., 2005). Moreover, personalized privacy preserving (Xiao and Tao, 2006) is also an important publishing scenario because publishers always ignore the concrete needs of individuals, which may cause serious privacy disclosures. Wang et al. (2009) propose the notion of FF-anonymity to eliminate the free-form attack for the microdata table, but they do not provide a specific anonymization algorithm to achieve FF-anonymity. Sei et al. (2019) assume all the QI values are potential private information and propose a general anonymization approach that can comply with frequency $(l_1, \cdots, l_q)$-diversity, entropy $(l_1, \cdots, l_q)$-diversity, and $(t_1, \cdots, t_q)$-closeness to protect all the values in the microdata table.

Secondly, an anonymity principle should be chosen or proposed to provide secure protections against privacy disclosures according to the assumed background knowledge and purpose of the adversary. Generally, the privacy threats can be divided into four categories, which are membership disclosure, identity disclosure, attribute disclosure, and probabilistic disclosure. Many useful anonymity principles can be used to prevent these privacy attacks. k-Anonymity (Sweeney, 2002) is one of the most powerful and widespread principles for protecting sensitive information, especially for defending identity attacks. l-Diversity (Machanavajjhala et al., 2006; Mehta and Rao, 2022) and t-closeness (Li et al., 2007) are popular and effective principles for preventing attribute disclosure. δ-Presence (Nergiz et al., 2007) and (d, γ)-privacy (Rastogi et al., 2007) hinder membership disclosure and probabilistic disclosure, respectively. Note that, according to Dwork (Dwork, 2006), absolute privacy protection is impossible. Therefore, a robust anonymity framework is supposed to prevent most leakages of confidential information by combining various principles. Additionally, differential privacy (Dwork and Roth, 2014), which is an effective model for protecting statistic data, does not need to assume the background knowledge of the adversary. It prevents different disclosures of queries by adding noise based on different mechanisms (Dwork et al., 2006; McSherry, 2010; Roth and Roughgarden, 2010; Geng and Viswanath, 2016; Liu, 2019).

Finally, an explicit algorithm should be presented that complies with at least one anonymity principle and minimizes information loss as far as possible. For example, slicing (Li et al., 2012) complying with l-diversity principle protects sensitive attributes, prevents attribute disclose and membership disclosure to some extent, and preserves better data utility than generalization. Microaggregation (Soria-Comas et al., 2016) is a flexible technique that satisfies k-anonymous t-closeness through merging clusters, and it can also adjust the priority of k-anonymity and t-closeness according to actual demands, such that microaggregation can choose to preserve more information utility by k-anonymity-first or provide more information security by t-closeness-first. Cross-bucket generalization (Li et al., 2018) complies with (k, l)-anonymity principle to prevent identity disclosure and attribute disclosure. It not only provides greater security for sensitive attributes but also preserves more information utility than generalization when the demand for attribute protection is higher than that for identity protection.

6. Conclusion and future study

This paper supposes that people can optionally set their sensitive values according to the personal requirements and proposes a novel technique, namely local generalization and bucketization, to provide secure protections for identities and sensitive values. The rationale is to divide the tuples into local equivalence groups and partition the sensitive values into local buckets through local generalization and local bucketization, respectively. Furthermore, the protections of local generalization and local bucketization are independent, so that their algorithms can be flexibly achieved according to practical requirements without weakening the other protection, respectively.

LGB is a flexible framework that can be used in many publishing scenarios for protecting confidential information. In the future study, we will combine the approach of online learning that automatically protects the incremental publishing data. Additionally, individual relationships, which are also considered as sensitive, are recommended to be studied in accordance with LGB.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aggarwal, C.C., 2005. On k-anonymity and the curse of dimensionality, very large data bases. 901–909.

Bayardo, R., Agrawal, R., 2005. Data privacy through optimal k-anonymization. In: 21st International Conference on Data Engineering (ICDE'05), pp. 217–228.

Byun, J.-W., Sohn, Y., Bertino, E., Li, N., 2006. Secure anonymization for incremental datasets. In: SDM 2006, pp. 48–63.

Dwork, C., Roth, A., 2014. The Algorithmic Foundations of Differential Privacy.

Dwork, C., 2006. Differential privacy. In: ICALP'06 Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II, pp. 1–12.

Dwork, C., Mcsherry, F., Nissim, K., Smith, A., 2006. Calibrating noise to sensitivity in private data analysis. Lect. Notes Comput. Sci., 265–284

Fung, B.C.M., Wang, K., Chen, R., Yu, P.S., 2010. Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Surv. 42 (4).

Geng, Q., Viswanath, P., 2016. The optimal noise-adding mechanism in differential privacy. IEEE Trans. Inf. Theory 62 (2), 925–951.

Kifer, D., Gehrke, J., 2006. Injecting utility into anonymized datasets. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 217–228.

Le, J., Zhang, D., Mu, N., Liao, X., Yang, F., 2018. Anonymous privacy preservation based on m-signature and fuzzy processing for real-time data release. IEEE Trans. Syst. Man Cybernet., 1–13

LeFevre, K., DeWitt, D., Ramakrishnan, R., 2006. Mondrian multidimensional k-anonymity. In: 22nd International Conference on Data Engineering (ICDE'06).

Li, T., Li, N., 2009. On the tradeoff between privacy and utility in data publishing. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517–526.

Li, N., Li, T., Venkatasubramanian, S., 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 106–115.

Li, T., Li, N., Zhang, J., Molloy, I., 2012. Slicing: A new approach for privacy preserving data publishing. IEEE Trans. Knowl. Data Eng. 24 (3), 561–574.

Li, B., Liu, Y., Han, X., Zhang, J., 2018. Cross-bucket generalization for information and privacy preservation. IEEE Trans. Knowl. Data Eng. 30 (3), 449–459.

Liu, F., 2019. Generalized gaussian mechanism for differential privacy. IEEE Trans. Knowl. Data Eng. 31 (4), 747–756.

Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M., 2006. L-diversity: privacy beyond k-anonymity. In: 22nd International Conference on Data Engineering (ICDE'06).

McSherry, F., 2010. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Commun. ACM 53 (9), 89–97.

Mehta, B.B., Rao, U.P., 2022. Improved l-diversity: Scalable anonymization approach for privacy preserving big data publishing. J. King Saud Univ. Comput. Inf. Sci. 34, 1423–1430.

Nergiz, M.E., Atzori, M., Clifton, C., 2007. Hiding the presence of individuals from shared databases. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 665–676.

Raju, N.V.S.L., Ramanath, M.N.S., Rao, P.S., 2022. An enhanced dynamic kc-slice model for privacy preserving data publishing with multiple sensitive attributes by inducing sensitivity. J. King Saud Univ. Comput. Inf. Sci. 34, 1394–1406.

Rastogi, V., Suciu, D., Hong, S., 2007. The boundary between privacy and utility in data publishing, very large data bases. 531–542.

Roth, A., Roughgarden, T., 2010. Interactive privacy via the median mechanism. In: Proceedings of the forty-second ACM Symposium on Theory of Computing, pp. 765–774.

Ruggles, S., Flood, S., Goeken, R., Grover, J., Meyer, E., Pacas, J., Sobek, M., 2020. IPUMS USA: Version 10.0 [dataset]. https://doi.org/10.18128/D010.V10.0.

Safi, S.M., Movaghar, A., Ghorbani, M., 2022. Privacy protection scheme for mobile social network. J. King Saud Univ. Comput. Inf. Sci. 34, 4062–4074.

Sei, Y., Okumura, H., Takenouchi, T., Ohsuga, A., 2019. Anonymization of sensitive quasi-identifiers for l-diversity and t-closeness. IEEE Trans. Dependable Secure Comput. 16, 580–593.

Soria-Comas, J., Domingo-Ferrer, J., Sanchez, D., Martinez, S., 2016. t-closeness through microaggregation: Strict privacy with enhanced utility preservation. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pp. 1464–1465.

Sweeney, L., 2002. Achieving k-anonymity privacy protection using generalization and suppression. Int. J. Uncert. Fuzziness Knowl.-Based Syst. 10 (5), 571–588.

Sweeney, L., 2002. k-anonymity: a model for protecting privacy. Int. J. Uncert. Fuzziness Knowl.-Based Syst. 10 (5), 557–570.

Wang, K., Fung, B.C.M., 2006. Anonymizing sequential releases, in. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 414–423.

Wang, K., Fung, B.C.M., Dong, G., 2005. Integrating private databases for data analysis, intelligence and security informatics. 171–182.

Wang, K., Xu, Y., Fu, A.W.-C., wing Wong, R.C., 2009. Ff-anonymity: When quasi-identifiers are missing. In: 2009 IEEE 25th International Conference on Data Engineering, pp. 1136–1139.

Xiao, X., Tao, Y., 2006. Anatomy: simple and effective privacy preservation, in. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 139–150.

Xiao, X., Tao, Y., 2006. Personalized privacy preservation. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 229–240.

Xiao, X., Tao, Y., 2007. M-invariance: towards privacy preserving re-publication of dynamic datasets, in. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 689–700.

Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.-C., 2006. Utility-based anonymization using local recoding, in. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–790.

Zhang, Q., Koudas, N., Srivastava, D., Yu, T., 2007. Aggregate query answering on anonymized tables. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 116–125.